

Remarks

These Remarks are in reply to the Office Action mailed May 30, 2008. Claims 50-63 and 71-75 were pending in the Application prior to the outstanding Office Action. Claim 75 is currently being amended. No claims are currently being canceled or added. Thus, claims 50-63 and 71-75 remain for the Examiner's consideration. In view of the following remarks, reconsideration and withdrawal of the outstanding rejections are respectfully requested.

I. Summary of Examiner's Rejections

Claims 50-63 and 71-75 were rejected under 35 U.S.C. § 103(a) for allegedly being unpatentable over Muller ("Event-Oriented Dynamic Adaption of Workflows: Model, Architecture, and Implementation") in view of Peltz ("Web Service Orchestration", HP, 1-2003).

II. Examiner Interview Summary

Applicant's undersigned representative thanks the Examiner for the telephonic interview that took place on August 19, 2008, between the Examiner and Applicant's representatives Karl Kenna (Reg. 45,445) and Jeffrey Kurin (Reg. 41,132).

During the Interview, Applicant's representatives explained why they believe that the inventions of independent claims 50 and 75 were not obvious based on Muller in view of Peltz.

Specifically, Applicant's representatives explained why they believe that Muller and Peltz do not teach a "Web Service source file [that] includes a source code and classes therein and **a workflow definition created using a workflow language that is specified in the form of annotations to the source code and the classes**, and wherein said workflow language extends the source code with a plurality of workflow constructs provided as XML commands that are then used as the annotations to the source code and the classes, including constructs for defining parallel processing of a workflow and separate workflow branches therein, and wherein the workflow definition further includes a construct to terminate the parallel processing of the workflow when certain conditions are met", as required by exemplary claim 75.

No agreement was reached during the Interview. Rather, the Examiner requested that Applicant submit a Reply that summarized the arguments discussed during the Interview, so that the Examiner can more fully consider such arguments.

III. Discussion of Claims

A. Claim 50

Claim 50 is reproduced below for the convenience of the Examiner.

50. A system for utilizing a workflow language, comprising:
- a computer including a processing device operating thereon;
 - a program source file stored on a computer readable medium, wherein the program source file includes a source code and classes therein and a workflow definition created using a workflow language that is specified in the form of annotations to the source code and the classes, and wherein said workflow language extends the source code with a plurality of workflow constructs, including constructs for defining parallel processing of a workflow and separate workflow branches therein, and wherein the workflow definition further includes a construct to terminate the parallel processing of the workflow when certain conditions are met; and
- means for creating a workflow program according to said workflow definition, including
- means for the computer to read the source file and process the plurality of workflow constructs to activate a workflow, including creating separate workflow processes corresponding to the separate workflow branches,
 - means for activating each of the separate workflow processes to subsequently generate activities at the computer as defined by each workflow branch, and
 - means for determining when the certain conditions specified in the source file have occurred and then terminating the parallel processing of the workflow. (emphasis added)

As is explained in the specification, an embodiment of Applicant's invention (including the embodiment of claim 50) enables developers that are used to using a specific programming language (e.g., JAVA) because they are familiar with that programming language, to extend the programming language by adding workflow constructs (e.g., parallelism) that are missing from the programming language (e.g., JAVA) but desirable. The system of claim 50, for example, can enable developers that are used to working with Java to add workflow definitions using a workflow language ***in the form of annotations to the source code and the classes*** of JAVA source code.

Such “annotations” would normally be ignored when the source code was processed, because “annotations” normally are not part of the program itself, and they normally have no direct effect on the operation of the source code they annotate. However, in this invention, such annotations (e.g., created using a workflow language, and provided, e.g., as XML commands) are read by the system and are used to create a workflow program, rather than being ignored. Claim 50 is limited to adding a specific workflow construct, which is parallel processing.

Applicant admits that workflow language and workflow constructs, including parallel processing, are not new. Applicant also admits that program source files that include source code with classes therein are not new. Nevertheless, Applicant believes that the invention of claim 50 provides a novel and unobvious way to use **annotations** to readily and easily add workflow processing (including parallel processing) to source code, such as a Java source file or a Web Service source code, that does not otherwise provide for such workflow processing.

In view of the above, and the discussion below, Applicant asserts that the cited references Muller and Peltz, alone or in combination, do not teach or suggest “a program source file [that] includes a source code and classes therein and a workflow definition created using a workflow language that is specified in the form of annotations to the source code and the classes, and wherein said workflow language extends the source code with a plurality of workflow constructs, including constructs for defining parallel processing of a workflow and separate workflow branches therein, and wherein the workflow definition further includes a construct to terminate the parallel processing of the workflow when certain conditions are met”, as required by claim 50.

Muller is related to an agent-based workflow management system that can deal with various failure situations. Figure 1-2 of Muller (on page 5 of Muller) illustrates a simplified architecture of a workflow management system, which includes a workflow definition. The Examiner admitted in the first full paragraph on Page 6 of the Office Action that “Muller does not explicitly address that the program source file includes a source code and classes therein, and a workflow definition created using a workflow language that is specified in the form of annotations to the source code and the classes.” Further, Applicant’s undersigned representative reviewed Muller to confirm that Muller indeed does not teach such features of claim 50.

In the second full paragraph on Page 6 of the Office Action, it was alleged that Peltz discloses the deficiencies of Muller. More specifically, it was asserted in the Office Action that certain portions of pages 5, 11 and 10 of Peltz disclosed “a program source file [that] includes a source code and classes therein and a workflow definition created using a workflow language that is specified in the form of annotations to the source code and the classes, and wherein said

workflow language extends the source code with a plurality of workflow constructs”. Applicant respectfully disagrees, as explained below.

The Office Action cited to the “early work” on page 5 of Peltz for disclosing that a workflow definition can be invoked using Web Services Flow Language (WSFL). The Office Action cited to the first paragraph on page 11 of Peltz for disclosing that certain process logic can be written using Java program source which includes program source code and classes. However, page 5 and 11 of Peltz do not discuss annotations to source code and classes. Thus, pages 5 and 11 of Peltz could not possibly teach that annotations to source code and classes can be used to add workflow constructs, in the manner required by claim 50.

In the second full paragraph of Page 6 of the Office Action, the Examiner also cites to page 10 of Peltz, which states that “Conversations within Workshop can be built with JavaDoc annotations tags. For example, to start a conversation, a developer would include the tag, “@jws:conversation: phase=start” (see page 10 of Peltz, 3rd full paragraph). In summary, this portion of Peltz appears to state that a conversation can be started using a JavaDoc annotation tag. However, a “conversation” is a way to identify a two-way communication, e.g., between one client application and one Web Service so that messages are always returned to the correct client. More specifically, a conversation is a mechanism that causes a response to a request to be returned to the correct requestor.

In contrast, a workflow is a set of steps or actions that occur in a particular order, and in response to specific conditions, in order to perform a task. Thus, there is a clear difference between a “conversation” and a “workflow”. Accordingly, Peltz’s statement that a JavaDoc annotation tag can be used to “start a conversation” does not teach or suggest that a workflow definition can be created using a workflow language that is specified in the form of annotations to source code and classes of a program source file, as is required by claim 50.

For at least the above reasons, Applicant respectfully asserts that claim 50 is not obvious in view of Muller and Peltz, alone or in combination. Accordingly, reconsideration and withdrawal of the 103(a) rejection of claim 50 is respectfully requested. In the event that the Examiner does not withdraw this rejection, Applicant respectfully requests that the Examiner explain in more detail how he is reading Peltz to teach the above mentioned deficiencies of Muller.

B. Claims 51-56

Claims 51-56 depend from and add additional features to claim 50. Applicant asserts that these claim are patentable for at least the reason that they depend from independent claim 50, as well as for the features that they add.

C. Claim 57

Claim 57 is reproduced below for the convenience of the Examiner.

57. A method for utilizing a workflow language, comprising:

selecting a program source file including a workflow definition created using a workflow language, wherein the program source file includes a source code and classes therein and a workflow definition created using a workflow language that is specified in the form of annotations to the source code and the classes, and wherein said workflow language extends the source code with a plurality of workflow constructs, including constructs for defining parallel processing of a workflow and separate workflow branches therein, and wherein the workflow definition further includes a construct to terminate the parallel processing of the workflow when certain conditions are met; and

using a workflow program according to said workflow definition, including

processing, using a computer including a processing device operating thereon, the plurality of workflow constructs to activate a workflow, including creating separate workflow processes corresponding to the separate workflow branches,

activating each of the separate workflow processes to subsequently generate activities at the computer as defined by each workflow branch, and

determining when the certain conditions specified in the source file have occurred and then terminating the parallel processing of the workflow.

Claim 57 includes the step of “selecting a program source file including a workflow definition created using a workflow language, wherein the program source file includes a source code and classes therein and **a workflow definition created using a workflow language that is specified in the form of annotations to the source code and the classes**, and wherein said workflow language extends the source code with a plurality of workflow constructs, including constructs for defining parallel processing of a workflow and separate workflow branches therein, and wherein the workflow definition further includes a construct to terminate the parallel processing of the workflow

when certain conditions are met” (emphasis added). For similar reasons to those discussed above with regards to claim 50, Applicant asserts that Muller and Peltz, alone or in combination, do not teach or suggest these features of claim 57. Accordingly, Applicant respectfully requests that the 103(a) rejection of claim 57 be reconsidered and withdrawn.

D. Claims 58-63

Claims 58-63 depend from and add additional features to claim 57. Applicant asserts that these claim are patentable for at least the reason that they depend from independent claim 57, as well as for the features that they add.

E. Claim 75

Claim 75 has been amended to further distinguish the claim from the Muller and Peltz. Claim 75, as amended, is reproduced below for the convenience of the Examiner.

75. A system for utilizing a workflow language, comprising:
- a computer including a processing device operating thereon;
 - a Web Service source file stored on a computer readable medium, wherein the Web Service source file includes a source code and classes therein and a workflow definition created using a workflow language that is specified in the form of annotations to the source code and the classes, and wherein said workflow language extends the source code with a plurality of workflow constructs provided as XML commands that are then used as the annotations to the source code and the classes, including constructs for defining parallel processing of a workflow and separate workflow branches therein, and wherein the workflow definition further includes a construct to terminate the parallel processing of the workflow when certain conditions are met;
 - a Work Flow file, separate from said Web Service source file, that includes workflow definition commands, and that are invoked by the Web Service source file using the workflow constructs; and
 - a logic on the computer for creating a workflow program according to said workflow definition, including reading both the Web Service source file and the Work Flow file, using the Web Service source file to create a Web Service running on the computer, and processing the plurality of workflow constructs to activate a workflow, including reading the definitions from the Work Flow file, creating separate workflow processes corresponding to the separate workflow branches, and

then activating each of the separate workflow processes to subsequently generate activities at the computer as defined by each workflow branch.

Claim 75 requires “a Web Service source file stored on a computer readable medium, wherein the Web Service source file includes a source code and classes therein and ***a workflow definition created using a workflow language that is specified in the form of annotations to the source code and the classes***, and wherein said workflow language extends the source code with a plurality of workflow constructs provided as XML commands that are then used as the annotations to the source code and the classes, including constructs for defining parallel processing of a workflow and separate workflow branches therein, and wherein the workflow definition further includes a construct to terminate the parallel processing of the workflow when certain conditions are met” (emphasis added). For similar reasons to those discussed above with regards to claim 50, Applicant asserts that Muller and Peltz, alone or in combination, do not teach or suggest these features of claim 75. Accordingly, Applicant respectfully requests that the 103(a) rejection of claim 75 be reconsidered and withdrawn.

Further, Applicant asserts that Muller and Peltz do not teach or suggest “a Work Flow file, separate from said Web Service source file, that includes workflow definition commands, and that are invoked by the Web Service source file using the workflow constructs”, as required by claim 75. The Office Action did not directly address this feature. Accordingly, if the Examiner is to again reject these claim, Applicant respectfully requests that the Examiner specify in more detail where he believes that this feature is taught by the applied references.

Further, Applicant asserts that Muller and Peltz clearly do not teach or suggest “a logic on the computer for creating a workflow program according to said workflow definition, including reading both the Web Service source file and the Work Flow file, using the Web Service source file to create a Web Service running on the computer, and processing the plurality of workflow constructs to activate a workflow, including reading the definitions from the Work Flow file, creating separate workflow processes corresponding to the separate workflow branches, and then activating each of the separate workflow processes to subsequently generate activities at the computer as defined by each workflow branch”, as required by claim 75. Without such logic the computer would not process the workflow definition created using a workflow language that is specified in the form of annotations to the source code and the classes, but would rather ignore such workflow definition.

The Office Action did not directly address this feature. Accordingly, if the Examiner is to again reject this claim, Applicant respectfully requests that the Examiner specify in more detail where he believes that this feature is taught by the applied references.

IV. Conclusion

In view of the above remarks, it is respectfully submitted that all of the claims now pending in the subject patent application should be allowable, and reconsideration thereof is respectfully requested. The Examiner is respectfully requested to telephone the undersigned if he can assist in any way in expediting issuance of a patent.

The Commissioner is authorized to charge any underpayment or credit any overpayment to Deposit Account No. 06-1325 for any matter in connection with this response, including any fee for extension of time, which may be required.

Respectfully submitted,

Date: August 29, 2008

By: /Jeffrey R. Kurin/
Jeffrey R. Kurin
Reg. No. 41,132

Customer No.: 23910
FLIESLER MEYER LLP
650 California Street, 14th Floor
San Francisco, California 94108
Telephone: (415) 362-3800